Table of Contents

[1 Document management history 5](#_Toc453952057)

[1.1 Prepared by 5](#_Toc453952058)

[1.2 Reviewed by 5](#_Toc453952059)

[1.3 Approved by 5](#_Toc453952060)

[1.4 Revision History 1](#_Toc453952061)

[2 Introduction 2](#_Toc453952062)

[3 Purpose of this document 3](#_Toc453952063)

[3.1 Project Scope 3](#_Toc453952064)

[3.2 Definitions, Acronyms and Abbreviations 3](#_Toc453952065)

[3.3 References 4](#_Toc453952066)

[3.4 Overview 4](#_Toc453952067)

[3.5 ~~Business Opportunity~~ 4](#_Toc453952068)

[3.6 ~~Assumptions~~ 4](#_Toc453952069)

[3.7 ~~Risks~~ 5](#_Toc453952070)

[4 Implementation 6](#_Toc453952071)

[4.1 Design Scope 6](#_Toc453952072)

[4.2 Dependencies/Impacts 6](#_Toc453952073)

[4.3 Development Tools 6](#_Toc453952074)

[4.4 Platforms 7](#_Toc453952075)

[4.5 Logical Architecture 8](#_Toc453952076)

[4.6 Architecture Decomposition 8](#_Toc453952077)

[5 Structural (Static Diagrams) 9](#_Toc453952078)

[5.1 Class Diagram 9](#_Toc453952079)

[5.2 Component Diagrams 10](#_Toc453952080)

[5.2.1 Mobile Client Application Components 10](#_Toc453952081)

[5.2.2 Mobile App native/Javascript components 11](#_Toc453952082)

[5.2.3 Mobile App HTML components 11](#_Toc453952083)

[5.2.4 Mobile App CSS components 11](#_Toc453952084)

[5.3 Activity/Behavioral Diagrams 12](#_Toc453952085)

[5.3.1 Overall (General) Application Process 12](#_Toc453952086)

[5.4 Sequence Diagrams – Rider/Driver 13](#_Toc453952087)

[5.4.1 Registration 13](#_Toc453952088)

[5.4.2 Login 13](#_Toc453952089)

[5.4.3 Accept EULA 13](#_Toc453952090)

[5.4.4 Profile 14](#_Toc453952091)

[5.4.5 Card info 14](#_Toc453952092)

[5.4.6 Last Trip 14](#_Toc453952093)

[5.4.7 Report a problem 14](#_Toc453952094)

[5.4.8 Contact 14](#_Toc453952095)

[5.5 Sequence Diagrams – Rider 14](#_Toc453952096)

[5.5.1 Request ride(now) 14](#_Toc453952097)

[5.5.2 Schedule ride 14](#_Toc453952098)

[5.6 Sequence Diagrams – Driver 15](#_Toc453952099)

[5.6.1 Vehicle details 15](#_Toc453952100)

[5.6.2 License details 15](#_Toc453952101)

[5.6.3 Insurance details 15](#_Toc453952102)

[5.6.4 Accept ride request 15](#_Toc453952103)

[5.6.5 Arrive Pickup location 15](#_Toc453952104)

[5.6.6 Start Ride 15](#_Toc453952105)

[5.6.7 End ride 15](#_Toc453952106)

[5.6.8 Go Online 16](#_Toc453952107)

[5.6.9 Go Offline 16](#_Toc453952108)

[5.6.10 Bank info 16](#_Toc453952109)

[5.6.11 Ride History 16](#_Toc453952110)

[5.6.12 Settings 16](#_Toc453952111)

[5.6.13 Request Ride 16](#_Toc453952112)

[5.6.14 On location change 16](#_Toc453952113)

[6 Layering 17](#_Toc453952114)

[6.1 Mobile App Presentation Layer 18](#_Toc453952115)

[6.1.1 Mobile Client Hybrid Application Screens 18](#_Toc453952116)

[6.2 Data Formats 24](#_Toc453952117)

[6.2.1 REST API data response 24](#_Toc453952118)

[Refer attached text doc 24](#_Toc453952119)

[7 Services 25](#_Toc453952120)

[7.1 Service Methods 25](#_Toc453952121)

[7.1.1 Service Methods1 25](#_Toc453952122)

[8 Security 26](#_Toc453952123)

[8.1 Authentication 26](#_Toc453952124)

[8.1.1 Mobile Authentication 26](#_Toc453952125)

[8.2 Authorization 26](#_Toc453952126)

[8.2.1 NA 26](#_Toc453952127)

[8.3 Database Connections 26](#_Toc453952128)

[8.4 Data protection 26](#_Toc453952129)

[8.5 Other considerations 26](#_Toc453952130)

[8.5.1 Session Management 26](#_Toc453952131)

[8.5.2 Cryptography 26](#_Toc453952132)

[8.5.3 Secure communications 26](#_Toc453952133)

[8.5.4 Data confidentiality 26](#_Toc453952134)

[8.5.5 System Design and Architecture 26](#_Toc453952135)

[8.5.6 System Configuration 26](#_Toc453952136)

[8.5.7 Secure Development 26](#_Toc453952137)

[9 Performance and Robustness 28](#_Toc453952138)

[9.1 Memory Management 28](#_Toc453952139)

[9.2 Processes and threads 28](#_Toc453952140)

[9.3 Concurrency 28](#_Toc453952141)

[9.4 Transactions 28](#_Toc453952142)

[9.5 Exception/Error handling 28](#_Toc453952143)

[9.6 Localization/Internationalization 28](#_Toc453952144)

[9.7 Benchmarking 28](#_Toc453952145)

[10 Configuration Management 29](#_Toc453952146)

[10.1 Source Control 29](#_Toc453952147)

[10.2 Configuration Files 29](#_Toc453952148)

[10.3 Initial Configuration 29](#_Toc453952149)

[10.3.1 Initial Packing of the Application 29](#_Toc453952150)

[10.4 Configuration Requirements 29](#_Toc453952151)

[NA 29](#_Toc453952152)

[10.5 Server Requirements 29](#_Toc453952153)

[11 Other Design Considerations 30](#_Toc453952154)

[11.1 Reporting 30](#_Toc453952155)

[11.2 Scheduling Impacts 30](#_Toc453952156)

[11.3 Non-Functional Requirements 30](#_Toc453952157)

[11.4 Backup 30](#_Toc453952158)

[11.5 Disaster Recovery 30](#_Toc453952159)

[11.6 Archive and Purge Strategy 30](#_Toc453952160)

[11.7 Installer 30](#_Toc453952161)

[11.8 Upgrade Strategy 30](#_Toc453952162)

[11.9 Data Conversion 30](#_Toc453952163)

[11.10 Metrics 30](#_Toc453952164)

[12 Appendix 31](#_Toc453952165)

# Document management history

## Prepared by

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Santhosh Joseph | Solution Architect | Intimation | 15/06/16 |
| Athira Satheesh | Developer | Intimation |  |
| Bibin Baby | Developer | Intimation |  |

## Reviewed by

The following section describes who has reviewed this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Shenjin Thomas | Client | BZRide inc | Xx/xx/xx |
|  |  |  |  |
|  |  |  |  |

## Approved by

The following section describes who has approved this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Shenjin Thomas | Client | BZRide inc | Xx/xx/xx |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 

## Revision History

The following section describes the revision history:

|  |  |  |
| --- | --- | --- |
| Version | Date | Comments |
| 0.1 | 15/06/2016 | Initial Draft |
| 1.0 | 24/09/2015 | Updated for different sections and diagrams |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

The introduction of the Solution Design Document provides an overview of the entire document. It includes the purpose, scope, definitions and references of this document.

# Purpose of this document

This document provides an overview of the BZRide Mobile Application. This document is intended to capture and convey the significant architectural and design decisions which have been made for designing and building the application. This is a technical reference document for team members involved in the development, testing and implementation of the application.

## Project Scope

BZRide is for driver and riders.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term / acronym / abbreviation | Definition |
| REST | Representational State Transfer |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| AJAX | Asynchronous JavaScript and XML |
| CSS | Cascaded Style Sheet |
| WCF | Windows Communication Foundation |
| EULA | End User License Agreement |

## References

This section provides a complete list of documents referenced in this document. Each document is identified by title, report number (if applicable), date, and publishing organization. Specify the sources from where the references can be obtained. This information may be provided by reference to an appendix or to another document.

This section contains a listing of existing documents that were referenced in order to prepare this document.

* High level Requirements document
* [SAD](https://share.ey.net/sites/tassdprojects/das/Shared%20Documents/Architecture/DAS%20Mobile%20App%20Solution%20Architecture%20Document.docx?Web=1)
* [Wireframe/VD](https://share.ey.net/sites/tassdprojects/das/Shared%20Documents/Architecture/Wireframes/DAS-v2.3.zip)
* Non Functional Requirements if any

## Overview

~~The App will have main features like view course details, submit posts and comments, book mark favorite posts, view announcements and notifications.~~

## ~~Business Opportunity~~

* ~~Better candidate retention and participation can be achieved by providing better learning experience to the candidates. Also with the usage of latest technology including mobile application support, better and longer candidate engagement can be ensured.~~
* ~~The proposed tool also opens up a quick, efficient and effective communication channel between the candidates, faculty and the INTIMATION Training Administrators.~~
* ~~Students can have a look at the upcoming course which will help them to plan for doing interested subjects and topics in addition to current course.~~
* ~~Discussion forums help candidates themselves to collaborate and brainstorm ideas, as well as discuss with faculty. Moderation in the forum can direct the discussions and can ensure that any kIntimation ideas / discussions gets captured and saved for future use.~~

## ~~Assumptions~~

~~For the purposes of designing and developing the BZRide Mobile application, the following assumptions have been made by the application development team:~~

* ~~The mobile client application would be developed as an Apache Cordova based hybrid application.~~
* ~~The application would be available only for iPhones.~~
* ~~The external users would be downloading the application from Apple App store.(TBD)~~
* ~~The internal users will be able to download the App from Citrix MAM Appstore.(TBD)~~
* ~~REST APIs would be available for accessing all data from Class Room Learning back end system.~~
* ~~App will support only online mode and there will not be any offline data store.~~

## ~~Risks~~

* ~~Information Security Clearance: Information Security and Risk assessment could highlight protection mechanism for authentication mechanism for users whose data that resides in CLS database system. However there is no confidential information as part of learning management system and authentication could be handled by custom web services.~~

# Implementation

## Design Scope

~~The BZRide Mobile application has been broken down into two major subsystems, the Apache Cordova and HTML5 based hybrid client application and a set of REST APIs that are running on CLS web portal.~~

## Dependencies/Impacts

Describe any dependencies and/or downstream impacts

## Development Tools

| Area of Development Use | Tool / Technology Currently Utilized |
| --- | --- |
| Integrated Development Environment | Dreamweaver, WebStorm, PHPStorm, XCode 7.0 |
| Development Framework | iOS SDK 9.0, Angular JS |
| Primary Development Languages | .NET C#, JavaScript, Objective C, HTML, CSS |
| Mobile Operating Systems | iOS 7.1 or Higher (iPhone) |
| Mobile Bus Width | iPhone 64 bit processing |
| User Experience Components | HTML5, CSS3 |
| Web Browser Platforms | WebKit enabled browsers |

## Platforms

Mobile App Development is planned in native languages and SDKs. Data will be pulled from SharePoint using REST API calls.

|  |  |
| --- | --- |
| iOS | |
| Company / developer: | Apple Inc. |
| OS family: | OS X |
| Widgets: | No, except in Notification Center |
| Initial release: | July 29, 2007 |
| Programmed in: | C, C++, Objective-C, |
| Easy media transfer: | With desktop application |
| Source model: | Closed, with open source components. |
| Open Source: | Kernel only |
| Customizability: | Limited unless jail broken |
| Available on: | iPod Touch, iPhone, iPad, Apple TV (2nd and 3rd generation) |
| Internet browsing: | Mobile Safari |
| Market share: | 14.9% of smartphones, 87% of tablets in North America (as of Jan'13) and 40.1% of tablets in Japan (as of Jan'13) |
| Available language(s): | 34 Languages |
| Latest stable release: | 9.0.2 for iPhones and iPads |

## Logical Architecture

The logical architecture of the solution is shown here:

## Architecture Decomposition



# Structural (Static Diagrams)

## Class Diagram

The Class Diagram of the solution is as shown below:

TODO

## Component Diagrams

### Mobile Client Application Components

Various application components of BZRide Mobile iPhone application are,

* HTML/CSS for rendering
* Javascript files for pulling data from back end. For above blocks corresponding HTML/CSS/JS files will be developed for pulling data from back end and presenting it for HTML files.

These sections are explained in detail in the Application Design section.

Other supporting components are explained below.

### Mobile App native/Javascript components

#### Constants.js

Shared variables and constant functions are defined in this JavaScript file.

#### Login.js

Login component performs the authentication with CLS back end with REST API.

#### Home.js

In the Home.js file there are 3 main operations are handled. Display my classes, display upcoming classes and display my posts.

#### myposts.js

This JavaScript file will helps application to populate posts made by logged in user. Call made to corresponding webservice.js function.

#### myclasses.js

list out call classes currently taken by the candidate. Call service js method.

#### postcomment.js

This class file helps post a comment to the discussion thread

#### favorites.js

Query all book marked posts by user. Call service js method.

#### Webservice.js

Makes the web service calls. Acts as a wrapper for other javascript files to get web service data.

This class can be further split into different service classes.

#### Utils.js

A common javascript class for holding common utility functions across all other classes.

### Mobile App HTML components

#### login.html

HTML code for display login page.

#### home.html

HTML code for display home page. Covers my classes and upcoming classes listing.

#### favorites.html

HTML code for display favorite posts.

### Mobile App CSS components

#### Home.css

Style sheet for home page.

#### login.css

Style sheet for login page.

## Activity/Behavioral Diagrams

### Overall (General) Application Process

The overall application process is depicted below.

Authentication Process

The User Authentication process of the application is as the following



## Sequence Diagrams – Rider/Driver

### Registration

Accept the details for registration on Application screen.

Accept credit/debit card info. Send card info to Stripe API(https://stripe.com/) for making it as an encrypted token

If rider pass all the details to web service method **RegisterRider** along with above card token.

If driver pass all the details to web service method **RegisterDriver** along with above card token.

### Login

Accept the login details on Application screen.

If rider Pass email and password to web service method **LoginRider**.

If driver Pass email and password to web service method **LoginDriver**.

### Accept EULA

After registration screen App moves to License screen.

On Accept call web service **AcceptEULA** and on server side mark LicenseAccepted flag as true.

On Reject show warning message “You cannot continue unless end user license agreement is accepted”

### Profile

Show Profile details on Application screen(**GetRiderProfile** or **GetDriverProfile** method). Edit the details if required. Once user click Save, call the web service method **UpdateDriverProfile** if driver. If rider call **UpdateRiderProfile**.

### Card info

Show card details on Application screen (**GetCardDetails** method). Edit the details if required. Once user click Save, invoke stripe API to get token and call the web service method **UpdateCardDetails** Method.

### Last Trip

Show very latest trip details by calling web service method **GetLastRideDetails.** If no data found show message “No Ride data found”

### Report a problem

Show an option text one paragraph for problem description and one title. On Send button save to server using service method **ReportProblem**

### Contact

Show contact Info as email and phone number. [info@bzride.com](mailto:info@bzride.com) and (xxx)-(xxx)-(xxxx)

## Sequence Diagrams – Rider

### Request ride(now)

Tap Request Now button from App screen.

Select Destination

Tap Confirm button.

Create a ride request with start location and destination location (lat/long) and address riderid etc.

Call web service method **CreateRideRequestImmediate**

### Schedule ride

Tap Request Now button from App screen.

Select Destination

Select Date and Time of travel (validate if it is greater than 2 days ahead of current date. Also warn if date is previous)

Tap Confirm button.

Create a ride request with start location and destination location (lat/long) trip date and time, and address riderid etc.

Call web service method **CreateRideRequestSchedule**

## Sequence Diagrams – Driver

### Vehicle details

The details will be included in **GetDriverProfile** service method as inside a tag vehicle

### License details

The details will be included in **GetDriverProfile** service method as inside a tag license

### Insurance details

The details will be included in **GetDriverProfile** service method as inside a tag insurance

### Accept ride request

The ride request is accepted as a push notification.

Handle the event in push notification handler for ios/android. Contains id, requestor name, start, destination

Call web service **ReadRideRequest**(id). If previous request is trying to be read show alert.

Driver accept it by tapping accept button within 20 seconds call web service method **AcceptRideRequest**(id,driverid)

Note: Back end service will monitor the accept status from any driver and if not accepted by anybody, give the message to requestor.

### Arrive Pickup location

The ride request is accepted as a push notification. Just drive to the pick location given by rider.

### Start Ride

The web service call **StartRide**(id) given by rider. Ride request table updated with status as riding-**R**

And driver table updated with status as driving-**D**. Open map view with start and end locations and get into start navigation mode as well. Use google maps view or waze View depending on settings menu option

### End ride

The web service call **EndRide**(id) given by rider. Ride request table updated with status as complete-**C**

And driver table updated with status as Available-**A.**

End navigation mode.

Calculate distance travelled.

Calculate time taken

### Go Online

The web service call **UpdateDriverAvailability**(id,true) to update status. Available-**A**

### Go Offline

The web service call **UpdateDriverAvailability**(id,false) to update status. Available-**O**

### Bank info

The **GetBankInfo** service method on updating details call method **UpdatebankInfo**

### Ride History

The **GetRideHistory** service method for pulling details

### Settings

Show a page with two radio buttons google map and waze and set the value G or W (1 or 2 also ) in local app settings

### Request Ride

Same work flow as Rider do Request

### On location change

On location change call service method **UpdateDriverLocation**(id,lat,long)

Layering

Layering represents an ordered grouping of functionality, with the application-specific located in the upper layers, functionality that spans application domains in the middle layers, and functionality specific to the deployment environment at the lower layers

## Mobile App Presentation Layer

### Mobile Client Hybrid Application Screens

#### Register User

#### C:\Users\Santhosh.joseph\Desktop\share\Projects\SanthoshJosephMobCoE\BZRIDE\Screens\User\Registeruser.png

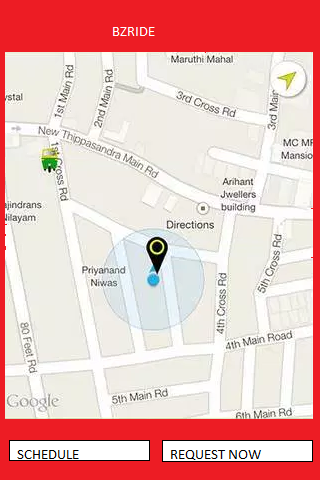
#### EULA Page

#### C:\Users\Santhosh.joseph\Desktop\share\Projects\SanthoshJosephMobCoE\BZRIDE\Screens\User\EULA.png

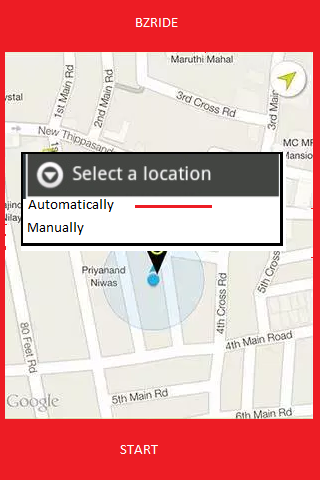
#### login Screen



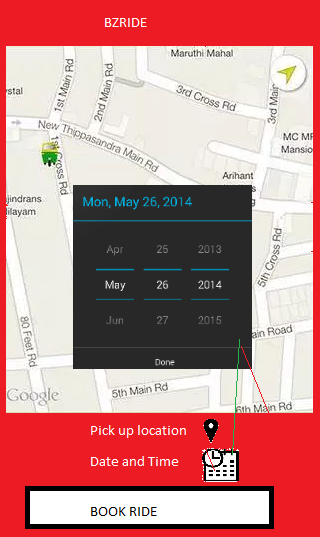
#### Home Page



#### Home Page-Pick Up location



#### Home Page-Schedule Trip



## Data Formats

### REST API data response

### Refer attached text doc

Services

SharePoint exposes REST services to GET/POST data to the SharePoint list. Major services are below

|  |  |
| --- | --- |
| Service URL | Feature Affected |
|  | Login |
|  | Logout |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Service Methods

sasasa

### Service Methods1

Security

## Authentication

The different Authentication models used are as follows:

### Mobile Authentication

## Authorization

### NA

## Database Connections

NA

## Data protection

NA

## Other considerations

### Session Management

### Cryptography

### Secure communications

1. All server communications are secured with SSL.??

### Data confidentiality

1. No data is stored on device and the App shows everything dynamically by calling REST APIs.
2. The course information/discussion threads are not of confidential nature.

### System Design and Architecture

### System Configuration

1. Mobile application uses REST API for data downloading.
2. Different class of users are given group permissions and thIntimation are classified as admin/faculty/student

### Secure Development

1. User credentials such as passwords, passphrases, PINs and digital certificates (PKI) are not stored anywhere in the mobile device.
2. Application does not allow sharing of its files and documents with unapproved Mobile Applications on the same mobile device.
3. Application restricts copy and paste operations so its clipboard information cannot be shared with unapproved Mobile Applications.
4. Application restricts its application data from being copied to connected computer-based or cloud-based backup services.

Performance and Robustness

## Memory Management

Application will load the data using REST API and it is kept in device memory for rendering the UI. Profiling using chrome can be done for better check on performance and memory management.

## Processes and threads

NA

## Concurrency

NA

## Transactions

## Exception/Error handling

Default exception/error handling methods provided by iOS/Android/Javascript will be used.

Details of the errors that the SharePoint application encounters are displayed to the user on the web page. The errors encountered are also logged.

## Localization/Internationalization

NA

## Benchmarking

NA

Configuration Management

For detailed information on management configuration, include section if applicable.

## Source Control

App source files are committed to GitHub.

## Configuration Files

NA

## Initial Configuration

### Initial Packing of the Application

The App will be bundled as ipa or apk file and uploaded to store.

## Configuration Requirements

## NA

## Server Requirements

NA.

# Other Design Considerations

These sections are optional and are included only if applicable.

## Reporting

NA

## Scheduling Impacts

NA

## Non-Functional Requirements

## Backup

NA

## Disaster Recovery

NA

## Archive and Purge Strategy

NA

## Installer

NA

## Upgrade Strategy

NA

## Data Conversion

NA

## Metrics

NA

# Appendix